

AMENDMENTS TO THE CLAIMS

The listing of claims below replaces all prior versions, and listings, of claims:

1 1. (Original) A database system comprising:
2 a persistent data storage device storing a first file management context and
3 having a pool of storage elements; and
4 a non-persistent memory storing a second file management context,
5 the first file management context to indicate allocated permanent files in
6 the pool of storage elements, and
7 the second file management context to indicate allocated temporary files
8 and permanent files in the pool of storage elements.

1 2. (Original) The database system of claim 1, wherein the first file
2 management context is a subset of the second file management context.

AI 1 3. (Original) The database system of claim 1, further comprising a control
2 module adapted to update an entry in the second file management context without
3 updating an entry in the first file management context to allocate a temporary file.

1 4. (Original) The database system of claim 3, wherein the control module is
2 adapted to update an entry in both the first and second file management contexts to
3 allocate a permanent file.

1 5. (Original) The database system of claim 1, wherein the pool of storage
2 elements comprises a pool of storage blocks.

1 6. (Original) The database system of claim 5, further comprising a control
2 module adapted to allocate one or more of the storage blocks to a temporary file or a
3 permanent file.

1 7. (Original) The database system of claim 5, wherein the first file
2 management context contains a first storage identifier map and a first allocation unit map,
3 the first storage identifier map indicating which storage identifiers have been allocated to
4 permanent files, and the first allocation unit map indicating which storage blocks have
5 been allocated to permanent files.

1 8. (Original) The database system of claim 7, wherein the second file
2 management context contains a second storage identifier map and a second allocation
3 unit map, the second storage identifier map indicating which storage identifiers have been
4 allocated to temporary and permanent files and the second allocation unit map indicating
5 which storage blocks have been allocated to temporary and permanent files.

1 9. (Original) The database system of claim 1, further comprising an access
2 module containing the non-persistent memory.

AI 1 10. (Original) The database system of claim 9, wherein the access module
2 comprises a data server to control access of the data storage device.

1 11. (Original) The database system of claim 10, further comprising an
2 application programming interface containing methods invocable by the data server to
3 access the first and second file management contexts.

1 12. (Original) The database system of claim 9, wherein the access module is
2 adapted to copy the first file management context from the persistent data storage device
3 to the non-persistent memory upon system restart.

1 13. (Original) The database system of claim 9, further comprising:
2 one or more other access modules;
3 one or more other persistent storage devices accessible by the
4 corresponding one or more other access modules; and
5 one or more other first and second file management contexts
6 corresponding to the one or more other access modules.

1 14. (Original) The database management system of claim 9, wherein the
2 access module performs at least one of a transaction locking and database logging
3 operation when updating the first file management context, and the access module is
4 adapted not to perform the transaction locking and database logging operations when
5 updating the second file management context but not updating the first file management
6 context.

1 15. (Original) The database management system of claim 1, wherein the
2 permanent files contain user data and the temporary files contain results of queries.

AI
1 16. (Original) A method for use in a database system having a persistent
2 storage device and a non-persistent memory, comprising:
3 storing a first file management context in the persistent storage device;
4 storing a second file management context in the non-persistent memory;
5 updating both the first and second file management contexts to allocate a
6 permanent file; and
7 updating the second file management context without updating the first
8 file management context to allocate a temporary file.

1 17. (Original) The method of claim 16, further comprising maintaining the
2 first file management context despite system reset, wherein the second file management
3 context is lost due to the system reset.

1 18. (Original) The method of claim 16, wherein the first file management
2 context contains a storage identifier map to allocate storage identifiers and an allocation
3 unit map to allocate blocks in the persistent storage device, and wherein updating the first
4 file management context comprises updating the storage identifier map and the allocation
5 unit map.

1 19. (Original) The method of claim 18, wherein the second file management
2 context contains a storage identifier map to allocate storage identifiers and an allocation
3 unit map to allocate blocks in the persistent storage device, and wherein updating the
4 second file management context comprises updating the storage identifier map and the
5 allocation unit map.

1 20. (Original) The method of claim 16, further comprising receiving a request,
2 the request containing a flag to indicate allocation of a temporary file or a permanent file,
3 wherein updating one or both of the first and second file management contexts is based
4 on the flag.

AI
1 21. (Original) The method of claim 16, further comprising copying the first
2 file management context to the non-persistent memory upon system startup.

1 22. (Original) The method of claim 16, further comprising performing at least
2 one of a transaction locking and database logging operation when updating the first file
3 management context and not performing the transaction locking or database logging
4 operation when updating the second file management context without updating the first
5 file management context.

1 23. (Original) An article comprising at least one storage medium containing
2 instructions that when executed cause a system to:
3 store a first file management context to indicate allocation of temporary
4 and permanent files; and
5 store a second file management context to indicate allocation of permanent
6 files.

1 24. (Original) The article of claim 23, wherein the instructions when executed
2 cause the system to further:
3 receive a request containing a flag to indicate a permanent file or a
4 temporary file;
5 update both the first and second file management contexts if the flag
6 indicates a permanent file; and
7 update the second file management context without updating the first file
8 management context if the flag indicates a temporary file.

1 25. (Original) The article of claim 24, wherein the instructions when executed
2 cause the system to update the first file management context by updating a first storage
3 identifier map and a first allocation unit map, and update the second file management
4 context by updating a second storage identifier map and a second allocation unit map.

1 26. (New) The database system of claim 1, further comprising a controller
2 adapted to:
3 perform at least one of a transaction locking and database logging
4 operation in response to detecting an update of the first file management context; and
5 not perform the transaction locking and database logging operations in
6 response to detecting an update of the second file management context without an update
7 of the first file management context.

1 27. (New) An article comprising at least one storage medium containing
2 instructions that when executed cause a system to:
3 store a first file management context in non-persistent memory to indicate
4 allocation of temporary and permanent files; and
5 store a second file management context in persistent storage to indicate
6 allocation of permanent files.

1 28. (New) The article of claim 27, wherein the instructions when executed
2 cause the system to:
3 update both the first and second file management contexts to allocate a
4 permanent file,
5 update the first file management context without updating the second file
6 management context to allocate a temporary file.

1 29. (New) The article of claim 28, wherein the instructions when executed
2 cause the system to:
3 perform at least one of a transaction locking and database logging
4 operation in response to detecting an update of the second file management context; and
5 not perform the transaction locking and database logging operations in
6 response to detecting an update of the first file management context without an update of
7 the second file management context.
